This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

A Utility–Based Incentive Scheme for P2P File Sharing in Mobile Ad Hoc Networks

Afzal Mawji and Hossam Hassanein
Telecommunications Research Lab, School of Computing
Queen's University
Kingston, Ontario, K7L 3N6, Canada
{mawji, hossam}@cs.queensu.ca

***Abstract*—A synergy exists between MANETs and P2P networks since both are decentralized and have dynamic topologies. This paper presents a utility–based incentive scheme for P2P file sharing over MANETs, designed to encourage users to contribute to the network by sharing files. The scheme combines a virtual pricing mechanism with file transfer delay estimation and allows users to choose the best peer to download from based on their preference for price vs. delay. Users can choose to lower the delay experienced or focus on obtaining files for the lowest price. Simulation results show that the utility–based scheme does a much better job of promoting sharing of files by users in comparison to an unpriced and a fixed price scheme, the delay per downloaded file tends to be lower, and the MANET has a greater network lifetime.**

I. INTRODUCTION

There is a synergy between mobile ad hoc networks (MANETs) and peer–to–peer (P2P) networks since both are decentralized networks in which nodes dynamically organize themselves, both must deal with frequent topology changes, and both attempt to be resilient to failure. Furthermore, MANET nodes perform the routing function themselves, as is the case with application–layer P2P overlays. Therefore it is natural to combine them together so that a P2P file sharing overlay runs on a MANET.

However, there are differences between P2P networks and MANETs. P2P networks are designed as overlays for deployment on the Internet, they tend to be very large–scale networks with many users on the "edge" of the Internet and the nodes have plenty of resources and generally do not move about. MANETs tend to have far fewer nodes, the links between them usually have a higher delay, the devices are energy–constrained, and the users are geographically nearby one another.

A common problem in P2P file sharing networks are *freeloaders*, users that download files for themselves, but do not share anything in return. Another undesirable is the *whitewasher*. Since the cost of creating an anonymous identity is free, whitewashers join the file sharing network, use up as much as they are permitted, leave, create a new identity, and then rejoin, repeating the cycle. To entice users to share files, many incentive schemes have been proposed. The idea behind such schemes is to encourage users to contribute files by rewarding users who do so and punishing those who do not. Selfish nodes are not exclusive to P2P systems. MANETs may also have nodes which do not forward traffic, thereby reducing the effectiveness and efficiency of the network as a whole. Such users require an incentive that encourages them to forward packets.

An effective incentive scheme for mobile ad hoc peer–to–peer networks (for which we use the abbreviation P2P–MANETs) would produce both effects: encourage users to forward traffic and also, for those participating in the file sharing network, to share files. The incentive scheme proposed here, to our knowledge the first proposed for these networks, attempts to do just that. MANET nodes that forward packets are rewarded with virtual credits for doing so. Nodes that are participating in the P2P file sharing system also earn credits for sharing files and forwarding traffic at the overlay level. Nodes that are part of the MANET but not currently part of the overlay may use the credits they have earned if they later decide to join the file sharing system. This allows nodes to be compensated for the cost of performing a service to the network. When a node forwards data or makes a file available for download, it is using its own resources, including CPU, bandwidth, and energy. The opportunity cost of forwarding data for other nodes is the use of one's own resources for the benefit of others because these resources are then unavailable for the node to use for its own purposes.

In addition to the credit–based scheme, our proposal introduces the idea of having each node indicate the delay it expects for the file transfer, so that a peer has an idea of the total delay required to obtain a file via a particular route. The peer then evaluates a utility function which considers both the credit cost and delay of obtaining the file from its peers and chooses the best one. The user is able to indicate a preference for lower cost or lower delay. Simulation results demonstrate that peers participating in the file sharing network are far more likely to share files when using the incentive scheme presented than they would for either a fixed price or unpriced scheme. Furthermore, the network is generally longer–lived when using the utility–based scheme. The next section examines several existing incentive schemes. Section 3 describes our incentive scheme in detail. Section 4 presents our simulation results, and Section 5 gives our conclusions.

II. EXISTING INCENTIVE SCHEMES

This section examines some existing incentive schemes. First we consider a scheme for P2P–MANETs, then we look

978-1-4244-2075-9/08/$25.00 ©2008 IEEE

at incentive schemes for MANETs only. Finally we look at incentive schemes for P2P networks only.

To our knowledge, there is only one incentive scheme for peer–to–peer overlays in MANETs. CHUM, proposed by Zhu and Mutka [1], is aimed at allowing MANET users to share access to Internet services, not file sharing networks. Users maintain trust values combining their own observations with recommendations from others. On top of this is a credit scheme in which nodes can borrow from one another. Nodes must maintain trust histories and credit limits for all other peers, which creates lots of data traffic and increases energy consumption.

There are two basic classes of incentive mechanisms in mobile ad hoc networks: reputation-based and credit-based. In reputation-based systems nodes earn a reputation based on past behavior. When they forward packets their reputation increases. When they don't forward packets, their reputation decreases. Nodes with a low reputation are avoided in path selection, and may be punished. To accomplish this task, nodes must monitor their neighbors and alert others when they feel a node is not cooperating [2]. Misbehaving nodes are avoided when selecting paths. Other schemes attempt to punish misbehaving nodes by not forwarding their traffic [3], [4]. These schemes require observations to periodically be sent network-wide, which may create excessive network traffic and increases energy expenditure. In credit-based schemes nodes earn some form of credit when they forward packets and spend credit to send their own data [5]. Some schemes, such as Sprite [6] and PIFA [7], require nodes to maintain receipts of packets sent and received and periodically send these to a centralized server which determines the amount of credit owed and deserved. Centralized servers are an impractical requirement in MANETs.

Most peer–to–peer network incentive systems are also based on either reputation– or credit–based schemes. Buchegger and Le Boudec [8] propose a system in which peers maintain both a reputation rating and a trust rating about all other peers they are interested in and periodically exchange this information, using a modified Bayesian approach to merge ratings. Liao et al. [9] propose that each byte uploaded gives a certain number of credits, each byte downloaded uses a certain number of credits, and each second online gives a certain number of credits. This encourages users to share files, and also to stay online.

III. A UTILITY–BASED INCENTIVE SCHEME

Several of the incentive schemes discussed in Section II do not consider file sharing, while others are designed for wired networks, and thus do not account for energy consumption, a vital consideration in MANET protocols. A P2P–MANET node should be able to decide to stop forwarding traffic because its battery level has fallen below a threshold and it wishes to conserve energy, or a node might decide that it will forward traffic only if it can charge a high price to compensate it for using highly limited bandwidth. Therefore, a P2P-MANET incentive protocol must consider a given device's capacity to serve the network, and allow it to establish a price that makes up for its costs.

A node in a P2P-MANET performs two main functions: from the point of view of the MANET, it can forward traffic. From the point of view of the P2P file sharing network, it can share its files. To encourage this, we propose a virtual credit–based system in which nodes earn credits when users download files from them, and spend credits when they in turn download files from others. To encourage nodes to forward traffic, nodes will also earn credits for relaying packets.

The price charged should reflect the demand on a node to forward data. When a node is being asked to forward a lot of data, this may quickly use up its resources. Therefore, it should charge more for them even if there is plenty left. This helps to prevent routes from becoming congested because central and popular routes will become overpriced and outlying routes will be more cost-effective. This also has the effect of being fairer to boundary nodes, who may not otherwise be able to earn credits for forwarding traffic.

It is undesirable for a node which has limited resources, such as little energy available, not enough bandwidth, or few spare CPU cycles, to forward traffic. Therefore the price that a node charges should also consider its available resources.

The utility–based incentive scheme presented in this paper is described in terms of an unstructured Gnutella–like network [10]. It is also perfectly compatible with structured networks.

Initially, when a peer joins the network it starts with a small number of credits proportional to the number of files it is sharing, and is not allowed to let this number fall below zero. Basing the starting credits on how many files the user shares encourages sharing and discourages freeloaders and whitewashers, while still allowing new nodes to begin downloading files.

Suppose peer c wants to obtain a particular file. It generates a query for that file (a *Query* message in Gnutella) and sends it to the peers it is connected to. The term client will be used for nodes that initiate queries and those that respond to queries will be called servers, even though nodes may be both clients and servers at the same time. This query is spread through the network by neighboring peers.

When a server receives the request and has the desired file it must determine a price for it. The price of the file is based on two factors: its perceived popularity, and its perceived availability. The more popular a file is the more can be charged for it. However, the more available a file is within the network the less its price should be in order to entice customers. File popularity is determined by the number of queries received for it in the past (the more queries received the more popular), while file availability is determined by the number of downloads in the past (the more downloads the less available it is). For example, if a peer has received many requests for a file but few downloads, its offering may be too expensive relative to other peers' price, so it may lower its price to encourage more downloads and thus earn more credits for itself.

The price charged by server s for file i is determined by the Cobb–Douglas [11] utility function $P_i^s = o_i^\gamma \times a_i^{1-\gamma}$. o_i is the perceived popularity of file i, calculated as an exponential moving average (EMA) of the number of requests for i, a_i is the perceived availability of i, calculated as an EMA of the number of downloads of i per request, and γ is a parameter that indicates s's preference for popularity against availability.

In addition to the price, s also calculates an estimated delay for serving the file. Since there is no historical information on which to determine arrival and service rates, for simplicity we assume constants for these values and thus use an M/M/1 queue. Therefore, the nodal delay is $d_s = \tau + T_x + D_e$ where τ is the wait time in the server's file sharing upload queue including service time, T_x is the file transmission time, and D_e is the delay due to energy constraints. This equation takes into consideration many of the peer's resources, namely CPU utilization, queue length, bandwidth, and energy.

From queueing theory, we know that $\tau = \frac{1/\mu}{1-\rho}$ where μ is the service rate of download requests and ρ is the utilization, equal to λ/μ where λ is the arrival rate of download requests. $T_x =$ file size/bandwidth.

The energy available at the server is an important consideration. If the node has little energy, it would be better off conserving it and not responding to requests. Therefore it indicates this preference by increasing the delay as energy decreases. D_e is defined as $D_e = \frac{k}{E+a}$ where E is the energy available at the time of the request. We define $a = \frac{D_{max} \times E_{min}}{E_{max} - E_{min}}$ and $k = \frac{E_{max} \times D_{max} \times E_{min}}{E_{max} - E_{min}}$, where D_{max} is the maximum delay, and E_{max} and E_{min} are the maximum and minimum energy levels, respectively. This ensures that when there is very little energy available, delay is very high, but this decreases quickly as energy increases. When energy is at a maximum, the added delay is zero. A query response message (*QueryHit* in Gnutella) is then sent back to client c, including the price and delay from the server.

Each node along the path back to c calculates its price of forwarding and estimated delay and appends this information to the response. It is in a node's interest to charge as much as it can in order to profit from providing the forwarding service. It is also in a node's interest to increase the delay as much as possible so that fewer resources are provided for other nodes, leaving more of its own resources for itself. However, the node must also be cautious not to set too high a price or delay, or the route will not be selected and the node will receive nothing.

An intermediate node will determine its estimated delay using the function $d_i = T_x + D_e$. The file is not placed on the intermediate node's file sharing upload queue, and it is assumed that packet queues introduce negligible delay so the τ component discussed previously is unused here. The file size is transmitted as part of the query response, so the intermediate node can calculate T_x from this value, and D_e is calculated as before.

The intermediate node determines its price for forwarding the data based on its perception of the demand for its services. If it has been asked to forward a lot of traffic in the recent past, then the chances are higher than it will be selected again. Servers do not need to add this extra profit component as the file they are serving already takes demand into account in the form of popularity and availability. The price that a node charges for forwarding traffic is the perceived demand calculated as an EMA of the amount of data forwarded, $p_i = \beta x_i + (1-\beta) p_{i-1}$, where x_i is the demand as of the ith packet, p_{i-1} is the historical price of the demand, and β is the smoothing constant.

When the query results reach c, they will contain the complete cost and delay of obtaining the file from that route, listed as a series of nodes with their associated prices and delay components. c can then choose the server s_i which gives the best value for its utility function, the Cobb–Douglas form equation $U_{s_i} = p_{s_i}^{\alpha} \times d_{s_i}^{1-\alpha}$. p_{s_i} is the sum of prices charged by all nodes en route from s_i to c, d_{s_i} is the sum of all delays en route from s_i to c, and α indicates the preference of c for price vs. delay. A value of $\alpha = 0$ indicates minimal delay no matter the cost, while $\alpha = 1$ indicates minimum price, no matter the delay. Additionally c must possess enough credits to obtain the file from the chosen server.

Suppose s_1 provides the best utility value and c has sufficient credits. Then c would send a *DLRequest* message to s_1 indicating its interest in obtaining the file, along with the required credits and the list of nodes and prices on the path. s_1 would remove its price from the total amount of credits sent by c, adding this to its own credits. Next, it would enqueue the file on its upload queue, associating the request from c with it in order to keep the credit information intact. When the file request reaches the head of the queue, it is sent to the next hop along with all remaining credit associated with the query. Each intermediate node along the path would then take the credit due it, and eventually the file will reach c with no remaining credit. Pseudo-code for the actions of each node involved in the transaction is given in Figure 1.

There may be nodes belonging to the underlying MANET which are not currently participating in the P2P overlay network. They may have been part of it in the past and/or may wish to join in the future. It is important that such nodes also obtain credit for forwarding traffic. Therefore, these nodes also determine price and delay information while forwarding queries responses. This requires a modification of the underlying routing protocol.

IV. SIMULATION RESULTS AND ANALYSIS

This section evaluates the performance of the utility–based incentive scheme for different price–delay preference (α) values and also compares it to two other schemes, one in which there are no prices, and one which prescribes fixed prices for files and forwarding.

A. Simulation Parameters and Metrics

The MANETs are simulated in *ns–2*, have an area of 1000m × 1000m, and contain 60 nodes. File queries are randomly generated at varying intervals, files are randomly placed throughout the network, the average file size is 5 MB, and a transmission rate of 54 Mbps is used. The random waypoint model is used for mobility and a uniform distribution with minimum speed of 1 m/s and maximum speed of 3 m/s is used for nodal velocity. The pause time is uniformly distributed with a mean of 60 seconds and the simulation time is set to 2 hours. DSR is used as the routing protocol and an unstructured Gnutella–like overlay with a maximum TTL of 7 is used. For the utility–based incentive scheme the following parameters

```
Server:

  receive_query:
    if not sharing file, ignore Query
    else {
      calculate price of uploading file based on
          popularity and availability,
              $P_i^s = o_i^\gamma \times a_i^{1-\gamma}$
      estimate download queue delay,
              $\tau = (1/\mu) \ / \ (1-\rho)$
      determine file transmission delay,
              $T_x$ = file size/bandwidth
      estimate equivalent delay due to energy
          constraints, $D_e = k/(E+a)$
      send QueryHit with price and delay
          information to next hop en route to client
    }

  receive_dl_req:
    place DLRequest on queue
    take credits

Intermediate Node:

  receive_query:
    decrement TTL
    if TTL > 0, forward Query to neighbors

  receive_queryhit:
    calculate price of forwarding file based on
        traffic demand, $p_i = \beta \ x_i + (1-\beta)p_{i-1}$
    determine file transmission delay,
            $T_x$ = file size/bandwidth
    estimate equivalent delay due to energy
        constraints, $D_e = k/(E+a)$
    forward QueryHit with new price and delay
        information to next hop en route to client

  receive_file_transfer:
    take credits
    send file to next hop en route to client

Client:

  generate_query:
    generate local query_id
    set TTL = Max_TTL
    send Query to all neighbors
    set timeout value for query_id

  receive_queryhit:
    place QueryHit in response queue

  timeout for query_id:
    set best_server = null
    for each QueryHit message in response queue {
      if query_id does not match, ignore
      calculate utility function for QueryHit
      if utility > best_server.utility and
              credits_available > QueryHit.cost
          set best_server = QueryHit.server
    }
    send DLRequest with credits to best_server
```

Fig. 1: Pseudo-code for server, intermediate nodes, and client

are used: D_{max} = 10 sec, ρ = query rate/avg. file transfer time, $\gamma = 0.5$, $\beta = 0.375$.

In each simulation experiment, the price–delay preference parameter (α) in the utility function is varied. In addition, simulations for overlays in which no pricing scheme is used and one which has a fixed price for downloads and forwarding packets were also performed in order to compare them with the flexible pricing scheme presented in this paper.

For the unpriced scheme, the download was obtained from the first node to respond to the query, which generally favors closer nodes. For the fixed price scheme, files were obtained from the cheapest source. Free-riders were not explicitly placed into any system, but as the results show they nevertheless made an appearance in the unpriced scheme. The network lifetime is defined as the time at which at least 80% of the nodes have used more than 90% of their energy.

TABLE I
ENERGY CONSUMPTION CONSTANTS USED IN SIMULATION

m_{send}	1.89	$mW \cdot sec/byte$
b_{send}	246	$mW \cdot sec$
m_{recv}	0.494	$mW \cdot sec/byte$
b_{recv}	56.1	$mW \cdot sec$
$b_{sendctl}$	120	$mW \cdot sec$
$b_{recvctl}$	29.0	$mW \cdot sec$

TABLE II
DOWNLOADS PER NODE COMPARISON

	Avg. DLs	Max. DLs	Min. DLs
$\alpha = 0.0$	27.0	41	22
$\alpha = 0.5$	24.3	34	17
$\alpha = 1.0$	21.7	30	11
Fixed price	17.0	35	6
Unpriced	22.3	44	29

The energy consumption model used in the simulations is the linear model proposed by Feeney [12]. Each MAC layer operation takes a certain amount of power as defined by $cost = m \times size + b$ where m is the incremental cost of the operation, b is the fixed cost, and $size$ is the amount of data sent or received. The constants are obtained by physical measurements for a Lucent IEEE 802.11 WaveLAN PC Card from [12] and are summarized in Table I.

B. Simulation Analysis

Promoting contribution by all users is the paramount goal of an incentive scheme. This can be determined by examining how much each user has contributed to the network. In particular, we can see how many files have been uploaded by users. Due to the random nature of both the file placement and the queries we cannot expect each user to contribute the same amount, however the closer they are the fairer the scheme tends to be.

Tables II and III provide the average number of downloaded and uploaded files per node for some preference values of the utility–based scheme as well as the unpriced and fixed price schemes for a 30 P2P node overlay simulation. The maximum and minimum values are also presented. Table III shows that the unpriced scheme has freeloading nodes even though none were explicitly placed into the system. One node alone accounted for nearly one–third of the total uploads in the scheme. By contrast, the fixed price scheme resulted in the fewest number of average downloads and yet the difference between maximum and minimum uploads is very high. The utility–based scheme shows that average downloads decreases as we focus on obtaining the lowest price, but in all cases the spread between minimum and maximum uploads is much lower than the other two systems.

Delay is an important metric for file sharing networks as users want downloaded files to arrive as soon as possible, but nodes would prefer to delay uploads so as to preserve bandwidth. Figure 2 shows the average delay per download for some preference values of the utility–based scheme as well

TABLE III
UPLOADS PER NODE COMPARISON

	Avg. ULs	Max. ULs	Min. ULs
$\alpha = 0.0$	27.0	39	22
$\alpha = 0.5$	24.3	34	14
$\alpha = 1.0$	21.7	28	10
Fixed price	17.0	69	2
Unpriced	22.3	207	0

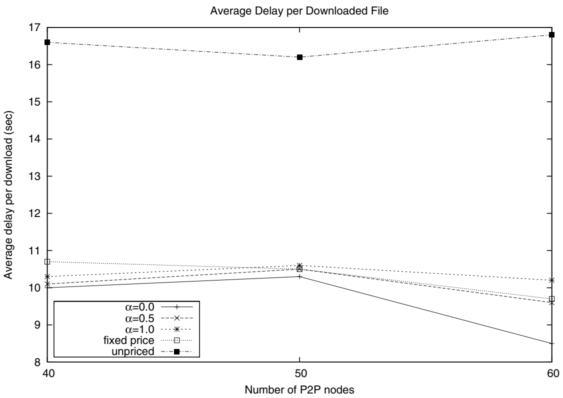

Fig. 2: Average delay per download

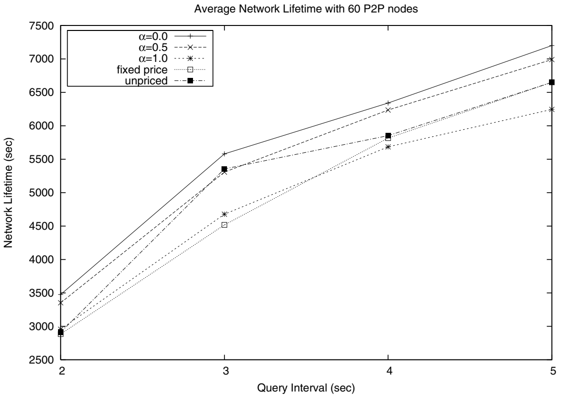

Fig. 3: Average Network Lifetime

as the unpriced and fixed price schemes for varying numbers of P2P nodes. Lower values of α indicate greater preference for the delay component (i.e. lower delay) at the expense of the price. The utility–based scheme shows the lowest delay, depending on the parameter value, and the unpriced scheme has a significantly higher average delay per downloaded file.

When a node has used all of its energy it can no longer participate in the network. Once a large number of nodes are in this situation the network is effectively useless. Therefore the greater the lifetime of the network, the more useful it is to its participants. Figure 3 shows the lifetime of the network for various parameter values of the utility–based scheme as well as the unpriced and fixed price schemes. Lower α values focus on reducing delay and tend to choose nearer servers, reducing energy consumption. Higher α values choose the lowest price regardless of delay and may therefore choose far away servers. The unpriced and fixed price schemes have fluctuating lifetimes, but tend to perform poorly in comparison.

V. CONCLUSIONS

This paper introduced a utility–based incentive scheme for P2P file sharing over MANETs. To our knowledge, this is the first such scheme proposed for these networks. Price was determined based on demand for a file and on intermediate nodes' costs. In addition, this paper presented the novel idea of estimating delay for file transfers and considering energy as part of this delay. Finally, the system allows the user to determine which server to download from based on a utility function that considers pricing and delay information.

Simulation results confirm that the scheme outlined achieves its goal of promoting cooperation amongst the overlay nodes even though each user is interested only in maximizing its own benefit. In comparison to unpriced and fixed price schemes, the files uploaded were more evenly spread among nodes, and we obtained a greater number of average downloads than the other two schemes. Furthermore, the lifetime of the network for the utility–based scheme can be increased by focusing on lower delay when choosing a server. Finally, the delay per downloaded file tends to be lowest for the utility–based scheme.

In the future, we will add mechanisms to prevent white-washers from lying about how many files they are initially sharing and falsely obtaining credits. We will also ensure that no peer cheats others by not paying when they download a file. Finally, we will look at the ideas of borrowing and lending credits, maximizing utility over the entire network as opposed to each user, and how a server can maximize its income.

REFERENCES

[1] D. Zhu and M. Mutka, "Promoting cooperation among strangers to access internet services from an ad hoc network," in *IEEE PERCOM*, 2004.

[2] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *ACM MobiCom*, Boston, MA, 2000, pp. 255–265.

[3] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the confidant protocol: Cooperation of nodes – fairness in dynamic ad-hoc networks," in *IEEE/ACM MobiHOC*, June 2002, pp. 226–236.

[4] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Joint Working Conference on Communications and Multimedia Security*, 2002, pp. 107–121.

[5] L. Buttyán and J.-P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *Mobile Networks and Applications*, vol. 8, no. 5, pp. 579–592, 2003.

[6] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *IEEE INFOCOM*, vol. 3, San Francisco, March 2003, pp. 1987–1997.

[7] Y. Yoo, S. Ahn, and D. P. Agrawal, "A credit-payment scheme for packet forwarding fairness in mobile ad hoc networks," in *IEEE International Conference on Communications (ICC)*, vol. 5, 2005, pp. 3005–3009.

[8] S. Buchegger and J.-Y. L. Boudec, "A robust reputation system for p2p and mobile ad-hoc networks," in *Workshop on the Economics of P2P Systems*, 2004.

[9] W.-C. Liao, F. Papadopoulos, and K. Psounis, "An efficient algorithm for resource sharing in peer-to-peer networks," in *Networking Technologies, Services, and Protocols*, ser. Lecture Notes in Computer Science, vol. 3976, 2006, pp. 592–605.

[10] The annotated gnutella protocol specification v0.4. [Online]. Available: http://rfc-gnutella.sourceforge.net/developer/stable/index.html

[11] H. Varian, *Intermediate Microeconomics: A Modern Approach, 6th ed.* W. .W. Norton & Company, 2003.

[12] L. M. Feeney, "An energy–consumption model for performance analysis of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 6, no. 3, pp. 239–250, 2001.